텐서플로우를 통한 데이터 예측

생성자	선 선우 박
생성 일시	@2024년 8월 26일 오전 12:13
태그	인프런 파이썬

텐서플로를 통한 자동차 연비 예측하기

1. 필요 도구 가져오기

```
# 데이터 분석을 위한 pandas, 시각화를 위한 seaborn 불러오기
import pandas as pd
import seaborn as sns
```

2. 데이터셋 로드

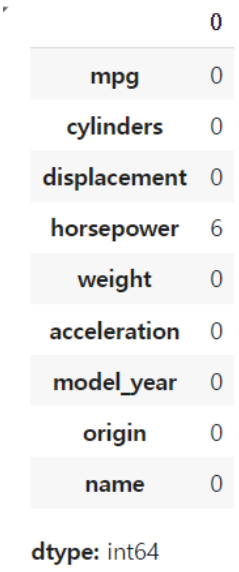
```
# 자동차 연비 데이터셋인 mpg 데이터셋 불러오기
df = sns.load_dataset('mpg')
df.shape
>> (398, 9)
```

- mpg 데이터셋을 df로 저장
- 행 398개, 열 9개

3. 결측치 확인

```
# 결측치의 합계 구하기
df.isnull().sum()
```

- horsepower에 결측치 6개



	0
mpg	0
cylinders	0
displacement	0
horsepower	6
weight	0
acceleration	0
model_year	0
origin	0
name	0

dtype: int64

4. 결측치 제거

```
#dropna로 결측치 제거
df = df.dropna()
df.shape
>> (392, 9)
```

- 결측치가 제거돼서 행의 수가 392개로 줄어듦

5. 수치 데이터만 가져오기

- 머신러닝이나 딥러닝 모델은 내부에서 수치계산을 하기 때문에 숫자가 아닌 데이터를 넣어주면 모델이 학습과 예측을 할 수 없음

```
# select_dtypes를 통해 object 타입을 제외하고 가져옴
# exclude 옵션 사용
df = df.select_dtypes(exclude="object")
```

6. 전체 데이터에 대한 기술 통계 확인

```
# describe를 통해 기술 통계값을 확인
# 수치형에 대해서만 요약
df.describe(include="all")
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year
count	392.000000	392.000000	392.000000	392.000000	392.000000	392.000000	392.000000
mean	23.445918	5.471939	194.411990	104.469388	2977.584184	15.541327	75.979592
std	7.805007	1.705783	104.644004	38.491160	849.402560	2.758864	3.683737
min	9.000000	3.000000	68.000000	46.000000	1613.000000	8.000000	70.000000
25%	17.000000	4.000000	105.000000	75.000000	2225.250000	13.775000	73.000000
50%	22.750000	4.000000	151.000000	93.500000	2803.500000	15.500000	76.000000
75%	29.000000	8.000000	275.750000	126.000000	3614.750000	17.025000	79.000000
max	46.600000	8.000000	455.000000	230.000000	5140.000000	24.800000	82.000000

7. 데이터셋 나누기
    - 전체 데이터프레임에서 df, train, test를 분리
    - trian_dataset : 학습에 사용 (예 : 기출문제)
    - test_dataset : 실제 예측에 사용 (예 : 실전문제)
    - 기출문제로 공부하고 실전 시험을 보는 과정과 유사
    - random_state
        - 데이터를 어떤 순서로 섞을 지
        - 다양한 모델을 사용해서 결과를 비교할 때 차이나지 않도록 도와줌
        - random_state = 42 → 42번째 데이터부터 시작해서 섞임

```
# 기존 392개의 80%인 314개를 뽑음
from inspect import trace
train_dataset = df.sample(frac = 0.8, random_state = 42)
train_dataset.shape
>> (314, 7)

# train_dataset에 들어가지 않은 값들을 test_dataset에 넣음
test_dataset = df.drop(train_dataset.index)
```

```
test_dataset.shape
>> (78, 7)
```

```
# train_dataset, test_dataset에서 label(정답)값을 꺼내 label을
# 문제에서 정답을 분리하는 과정
# train_labels : train_dataset에서 정답을 꺼내서 분리
# test_labels : test_dataset에서 정답을 꺼내서 분리

train_labels = train_dataset.pop('mpg')
train_labels.shape
>> (314,)

test_labels = test_dataset.pop('mpg')
test_labels.shape
>> (78, )

# pop을 사용해서 하나씩 줄어들어 있음
train_dataset.shape, test_dataset.shape
>> ((314, 6), (78, 6))
```

```
# mpg 제외 나머지 컬럼들 출력, 이 값들로 mpg 값 예측
train_dataset.head(2)
```

	cylinders	displacement	horsepower	weight	acceleration	model_year
79	4	96.0	69.0	2189	18.0	72
276	4	121.0	115.0	2795	15.7	78

```
# mpg 값 출력
train_labels.head(2)
```

	mpg
79	26.0
276	21.6

dtype: float64